



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение высшего
образования

"МИРЭА - Российский технологический университет"

РТУ МИРЭА

Институт информационных технологий (ИТ)
Кафедра математического обеспечения и стандартизации информационных
технологий (МОСИТ)

ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №5_1

по дисциплине
«Структуры и алгоритмы обработки данных»

Тема. Бит. Опер.

Выполнил студент группы ИКБО-42-23

Голев С.С.

Принял старший преподаватель

Муравьёва Е.А.

Москва 2024

ОГЛАВЛЕНИЕ

ЦЕЛЬ РАБОТЫ	3
1. УСЛОВИЯ ЗАДАЧ	4
2. ОТЧЁТ ПО ЗАДАНИЮ 1.....	5
2.1. Код используемый в программе	5
2.2. Результаты тестирования	8
3. ОТЧЁТ ПО ЗАДАНИЮ 2.....	10
3.1. Код используемый в программе	10
3.2. Результаты тестирования	11
4. ОТЧЁТ ПО ЗАДАНИЮ 3.....	12
4.1. Код используемый в программе	12
4.2. Результаты тестирования	12
4. ИСПОЛЬЗУЕМЫЕ ИСТОЧНИКИ	14

ЦЕЛЬ РАБОТЫ

Освоить приёмы работы с битовым представлением беззнаковых целых чисел, реализовать эффективный алгоритм сортировки на основе битового массива.

1.УСЛОВИЯ ЗАДАЧ

Задание 1:

Выполнить упражнения по применению битовых операций по изменению значений битов в ячейке оперативной памяти, созданию маски для изменения значения ячейки.

1 Номер бита	2 Номер бита	3 Множитель	4 Делитель	5 Задание для выражения
1-ый, 6-ой, 9-ый	1-ый, 6-ой, 9-ый	8	8	Установить n-ый бит в 1, используя маску

Задание 2:

Реализуйте вышеописанный пример с вводом произвольного набора до 8-ми чисел (со значениями от 0 до 7) и его сортировкой битовым массивом в виде числа типа unsigned char. Проверьте работу программы.

Задание 3:

Входные данные: файл, содержащий не более $n=107$ неотрицательных целых чисел, среди них нет повторяющихся.

Результат: упорядоченная по возрастанию последовательность исходных чисел в выходном файле.

2. ОТЧЁТ ПО ЗАДАНИЮ 1

2.1. Код используемый в программе

Опишем функции, используемые в программе для решения задач.

```
void task_1()
{
    unsigned int x = 0;
    unsigned int maska = (1 << 0) + (1 << 5) + (1 << 8);

    std::cout << "Введите искомую переменную:";
    std::cin >> x;
    std::cout << std::endl;

    unsigned int result_t1 = (x | maska);
    std::cout << "Здание 1" << std::endl;
    std::cout << "Изменяемое значение (x)\t" << x << "\t\t" << std::bitset<n>(x) << std::endl;
    std::cout << "Маска (maska)\t\t" << maska << "\t\t" << std::bitset<n>(maska) << std::endl;
    std::cout << "Результат операции\t" << result_t1 << "\t\t" << std::bitset<n>(result_t1) <<
std::endl;
    std::cout << std::endl;
}
```

Рисунок 2.1 – Функция для выполнения первой части первого задания варианта

Данная функция реализует изменение битов у исходного числа, с помощью маске, согласно варианту.

```
void task_2()
{
    unsigned int x = 0;
    unsigned int maska = pow(2,32) - 1;
    maska = maska - ((1 << 0) + (1 << 5) + (1 << 8));

    std::cout << "Введите искомую переменную:";
    std::cin >> x;
    std::cout << std::endl;

    unsigned int result_t1 = (x & maska);
    std::cout << "Здание 1" << std::endl;
    std::cout << "Изменяемое значение (x)\t" << x << "\t\t" << std::bitset<n>(x) << std::endl;
    std::cout << "Маска (maska)\t\t" << maska << "\t\t" << std::bitset<n>(maska) << std::endl;
    std::cout << "Результат операции\t" << result_t1 << "\t\t" << std::bitset<n>(result_t1) <<
std::endl;
    std::cout << std::endl;
}
```

Рисунок 2.1.1 – Функция для выполнения второй части первого задания
варианта

Данная функция реализует изменение битов на ноль у исходного числа, с помощью маски, согласно варианту.

```
void task_3()
{
    unsigned int x = 0;

    std::cout << "Введите искомую переменную:";
    std::cin >> x;
    std::cout << std::endl;

    unsigned result_t3 = (x << 3);
    std::cout << "Здание 3" << std::endl;
    std::cout << "Изменяемое значение (x)\t" << x << "\t\t" << std::bitset<n>(x) << std::endl;
    std::cout << "Множитель\t" << 8 << "\t\t" << std::bitset<n>(8) << std::endl;
    std::cout << "Результат операции\t" << result_t3 << "\t\t" << std::bitset<n>(result_t3) <<
    std::endl;
    std::cout << std::endl;
}
```

Рисунок 2.2 – Функция для выполнения третьей части первого задания варианта

Данная функция реализует умножение исходного числа на восемь, с помощью битовых операций, сдвиг в правую часть на 3 бита.

```
void task_4()
{
    unsigned int x = 0;

    std::cout << "Введите искомую переменную:";
    std::cin >> x;
    std::cout << std::endl;

    unsigned result_t4 = (x >> 3);
    std::cout << "Здание 4" << std::endl;
    std::cout << "Изменяемое значение (x)\t" << x << "\t\t" << std::bitset<n>(x) << std::endl;
    std::cout << "Делитель\t" << 8 << "\t\t" << std::bitset<n>(8) << std::endl;
    std::cout << "Результат операции\t" << result_t4 << "\t\t" << std::bitset<n>(result_t4) <<
    std::endl;
    std::cout << std::endl;
}
```

Рисунок 2.3 – Функция для выполнения четвертой части первого задания варианта

Данная функция реализует деление исходного числа на восемь, с помощью битовых операций, сдвиг в левую часть на 3 бита.

```

void task_5()
{
    unsigned int x = 0;

    std::cout << "Введите искомую переменную: ";
    std::cin >> x;
    std::cout << std::endl;

    unsigned int change = 0;
    unsigned int inv = 0;
    unsigned int maska = 0;
    unsigned result_t5 = 0;

    std::cout << "Введите число изменяемых битов:";
    std::cin >> change;
    std::cout << std::endl;
    std::cout << "Введите номера инвертируемых битов:" << std::endl;
    for (int i = 0; i < change; i++)
    {
        std::cin >> inv;
        if (inv < 0 || inv > 32)
        {
            std::cout << "Некорректный ввод";
            i--;
            break;
        }
        maska += (1 << inv);
    }

    for (int i = 0; i < n; i++)
    {
        if (((x >> i) & 1))
        {
            result_t5 += 1 << i;
        }
        else if (((maska >> i) & 1))
        {
            result_t5 += 1 << i;
        }
    }

    std::cout << std::endl;
    std::cout << "Изменяемое значение (x)\t" << x << "\t\t" << std::bitset<n>(x) << std::endl;
    std::cout << "Маска (maska)\t\t" << maska << "\t\t" << std::bitset<n>(maska) << std::endl;
    std::cout << "Результат операции\t" << result_t5 << "\t\t" << std::bitset<n>(result_t5) <<
    std::endl;
}

```

Рисунок 2.4 – Функция для выполнения пятой части первого задания варианта

Данная функция реализует установку 1 в указанные пользователем биты.

2.2. Результаты тестирования

Представим примеры работы программы.

[illegible]

Рисунок 2.5 – Тестирование функция первой и второй части первого задания
варианта

```
Введите номер задания: 3
Введите искомую переменную: 563

Здание 3
Изменяемое значение (x) 563 | 0000000000000000000000001000110011
Множитель 8 | 000000000000000000000000000000001000
Результат операции 4504 | 0000000000000000000000001000110011000
```

Рисунок 2.6 – Тестирование функция третьей части первого задания варианта

```
Введите номер задания: 4
Введите искомую переменную: 893

Здание 4
Изменяемое значение (x) 893 | 0000000000000000000000001101111101
Делитель 8 | 00000000000000000000000000000001000
Результат операции 111 | 0000000000000000000000000000001101111
```

Рисунок 2.7 – Тестирование функция четвертой части первого задания варианта


```

Введите номер задания: 5

Введите искомую переменную: 1589

Введите число изменяемых битов:10

Введите номера инвертируемых битов:
0
1
3
5
7
9
11
13
15
17

Изменяемое значение (x) 1589      |      00000000000000000000000011000110101
Маска (maska)              174763  |      00000000000000001010101010101011
Результат операции         175807  |      000000000000000010101011101011111

```

Рисунок 2.8 – Тестирование функция пятой части первого задания варианта

3. ОТЧЁТ ПО ЗАДАНИЮ 2

3.1. Код используемый в программе

Опишем функции, используемые в программе для решения задач.

```
void task_2()
{
    unsigned char a = 0;
    std::map<int, int> m;
    int size_input;
    int input;
    std::cout << "Введите количество чисел" << std::endl;
    std::cin >> size_input;

    std::cout << "Введите "<< size_input<<" чисел" << std::endl;
    for (int i = 0; i < size_input; i++)
    {
        std::cin >> input;
        if (m[input] == 0)
            a += 1<<input;
        m[input] += 1;
    }
    std::cout << "Отсортированный массив: " << std::endl;
    std::cout << std::bitset<8>(a) << std::endl;
    for (int i = 0; i < 8; i++)
    {
        if ((a>>i) & 1)
        {
            for (int j = 0; j < m[i] ; j++)
            {
                std::cout << i << ' ';
            }
        }
    }
}
```

Рисунок 3.1 – Функция, реализующая второе задание

В данной функции представлен алгоритм сортировки с использованием переменной char, также используется структура map для вывода повторяющихся переменных.

3.2. Результаты тестирования

Представим примеры работы программы.

```
Введите номер задания: 2
Введите количество чисел
8
Введите 8 чисел
2
4
5
3
3
7
6
1
Отсортированный массив:
11111110
1 2 3 3 4 5 6 7
```

Рисунок 3.2 – Тестирование функции, реализующей второе задание

4. ОТЧЁТ ПО ЗАДАНИЮ 3

4.1. Код используемый в программе

Опишем функции, используемые в программе для решения задач.

```
void task_3()
{
    std::bitset<size>* numbers = new std::bitset<size>;

    unsigned long int input;

    std::cout << "Введите по очереди семизначные числа (0 для завершения):" << std::endl;
    while (true) {
        std::cin >> input;
        if (input == 0) {
            break;
        }
        if (input >= size / 10 && input < size) {
            numbers->set(input - 1, 1);
        }
        else {
            std::cout << "Только семизначные числа!" << std::endl;
        }
    }

    std::ofstream outfile("sortedList.txt");
    for (unsigned long int i = 0; i < size; i++) {
        if (numbers->test(i)) {
            outfile << i + 1 << std::endl;
        }
    }
    outfile.close();
    std::cout << "Результат сортировки был записан в файл." << std::endl;
}
```

Рисунок 4.1 – Функция, реализующая третье задание

Данная функция представляет собой улучшенный алгоритм из задания два, который может принимать не только числа от 0 до 7.

4.2. Результаты тестирования

Представим примеры работы программы.

```
Введите номер задания: 3
Введите по очереди семизначные числа (0 для завершения):
1537598
4848266
1548484
4546465
8485955
6551511
5485451
2184656
2546598
0
Результат сортировки был записан в файл.
```

Рисунок 4.2 – Тестирование функции, реализующей третье задание

```
1537598
1548484
2184656
2546598
4546465
4848266
5485451
6551511
8485955

```

Рисунок 4.3 – Содержание файла, в который записан результат

4.ИСПОЛЬЗУЕМЫЕ ИСТОЧНИКИ

1. Лекции по Структуры и алгоритмы обработки данных / Рысин М. Л.
Москва, МИРЭА — Российский технологический университет.
2. Материалы по дисциплине Структуры и алгоритмы обработки данных /
Скворцова Л. А. Москва, МИРЭА — Российский технологический университет.